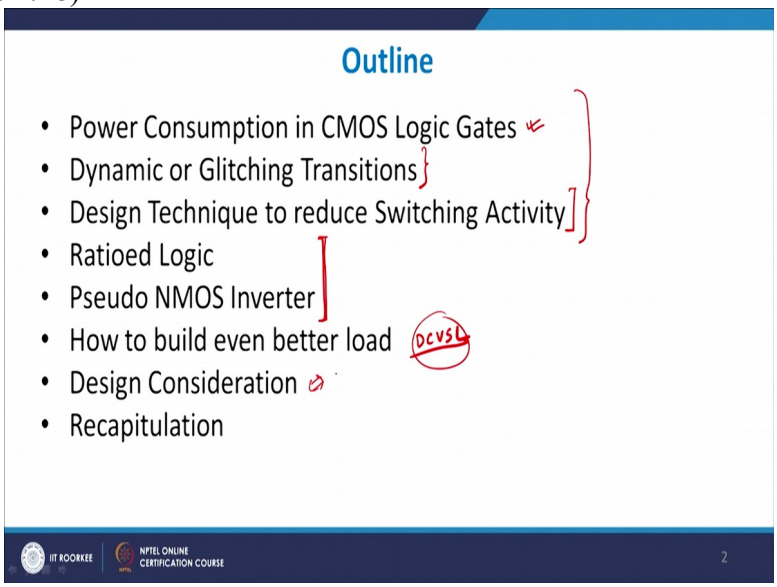
Microelectronics: Devices to Circuits
Prof. Sudeb Dasgupta
Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee
Module 11
Lecture 54
Combinational Logic Design -II

Hello, welcome to the next edition of the NPTEL online certification course on microelectronics devices to circuits. We start with the 2nd module of combinational logical design. In our previous module, we have seen what do you mean by propagation delay and how can you reduce your propagation delay in a complementary logic. We also saw what is the meaning of combinational logic and how can you design in a combinational logic, given a Boolean expression.

So if there is a Boolean expression available to you, how can you design it on a complementary logic? What we will be doing now is look into another important parameter and that is known as power dissipation. Let me give you the outline of what we are going to do.

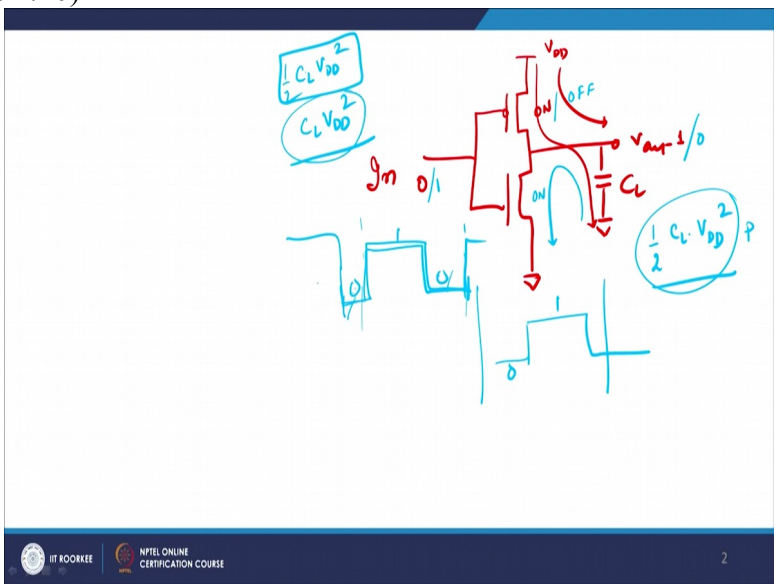
(Refer Slide Time: 1:18)



So we will look into the concept of power consumption and dissipation in CMOS logic gate. We will look into dynamic or glitching transitions and then how we can do, reduce the switching activity. And after we have understood all these basic concepts, we will take up what is known as a ratioed logic and within which we will study pseudo NMOS inverter.

And then we will look into what is known as a DCVSL which is also a better form of a ratioed logic. We will see how it works out. And then therefore, using DCVSL what are the design considerations for a DCVSL logic and then we will finally recapitulate for this particular module.

(Refer Slide Time: 2:10)



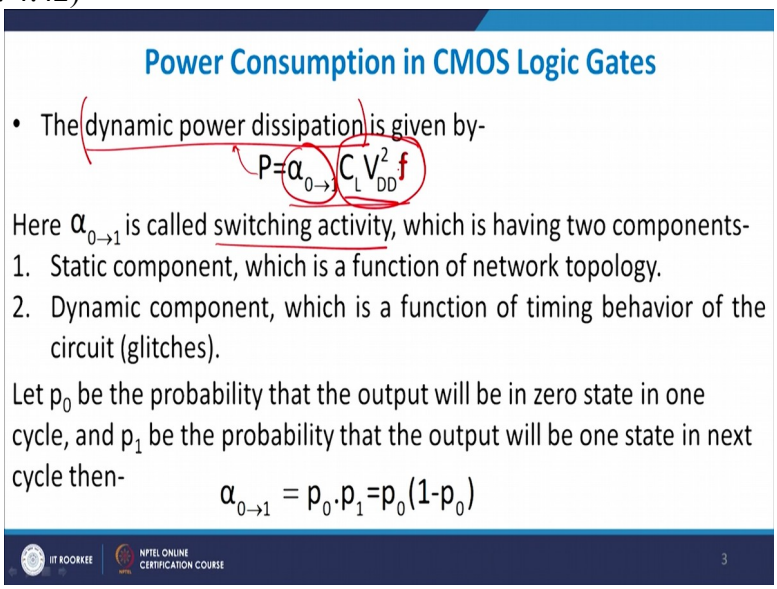
Let me explain to you before I move any forward, let me give you an idea about what is power consumption because we need to find out from basic this thing, inverter. You remember, if in an inverter, you have an input here right and then you have an output V_{out}, there is a load capacitance here. So whenever you have 0, this switches on as I discussed. Therefore this goes to 1 because your V_{DD} tries to charge. So you have this charging path in this manner. So C_L gets charged to 1.

Whenever now, whenever your input becomes equals to 1, this becomes off and this becomes on right? And when this becomes on, this charge which was there across C_L, this charges and this becomes 0. So this is the basic, fundamental principle of this inverter. Now you see, whenever your input is 0, you are drawing power from the V_{DD} rail and therefore, you have half C_L into V_{DD} square as the total amount of power which you are actually extracting from the V_{DD} rail over a single cycle.

So I am doing say, a 0 to 1 transition and that is all. I am doing a 0 to 1 transition, right. Now in the next half cycle, when 1 comes into picture, this switches on and the same charge or the energy or the power actually goes down to the ground. So over a cycle of, over a period of cycle or over one period of cycle, 0 to 1 transition, you actually utilized C_L times V_{DD} square. Half plus half because half you have taken and half you have done but then total will be, total power dissipation will be still equals to half C_L V_{DD} square, right.

Now with this knowledge, with this basic idea, we can say that the total power consumption over say one cycle of operation, so 0 1 0, so this is one cycle of operation. We have 2 times 0 coming into picture. So half plus half is equal to C_L into V_{DD} square, right. So over 1 period of cycle from this to this, this is one period of cycle right. So you are starting from here, ending here, you have got 0 0 and therefore we get C_L V_{DD} square as the power consumed from the V_{DD} rail.

(Refer Slide Time: 4:42)

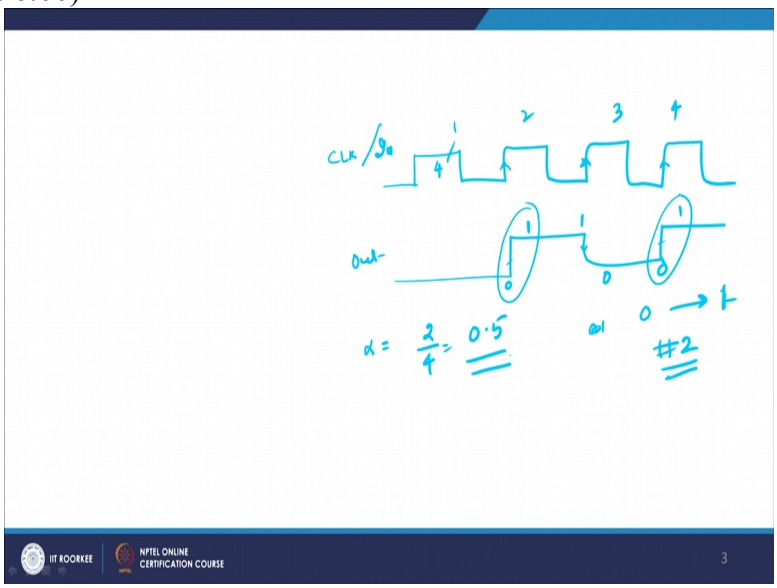


With this knowledge, what we do is we define another term known as dynamic power dissipation. It primarily means that, whenever the inverter is in operation mode, operation in the sense that whenever your input is varying from 0 to 1 and 1 to 0 and there is a large input bit train coming into picture and the output is also therefore varying from 0 to 1 and 1 to 0 we define this to be as a dynamic operation. What is a static operation? Static means that when it is given 1

or 0 and output is latched to either 0 or 1 and you have fixed that operation, we define that to be a static operation.

Then we define dynamic power dissipation as P equals, P is the dynamic power dissipation. It is given as this quantity, where alpha is defined as a switching activity or switching parameter. And C_L V_{DD} square we have already discussed is the power multiplied by frequency, you can understand why. Because in one cycle, you are dissipating C_L V_{DD} square. So if there are f such cycles available to you, this will be the total power which you will be dissipating in f such cycles. So that is the reason, you multiply with f. An interesting term comes here which is 0 to 1, alpha 0 to 1. I will explain this to you in a detailed fashion.

(Refer Slide Time: 6:00)

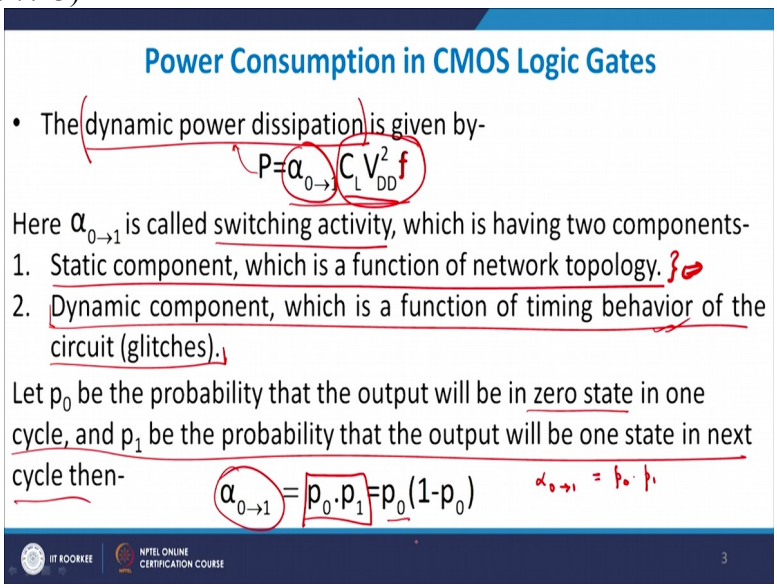


Let us see, let us say you have a clock in this manner right? You have a clock and there are 4 clock cycles which are there. Then let us suppose my output goes something like this. This is my clock or input, let us suppose and this is my output. So output I have got, so I have got something and then at this rising edge of the clock, I have a 0 to 1 transition. Similarly, at this rising edge of the clock, I have a 0 to 1 transition. So please understand, at this edge, at this rising edge of the clock I have 1 to 0 transition, right?

So please understand 0 to 1 transitions are power consuming and power dissipating sort of cycles, right. So you see, of the 4 clock cycles here, 1, 2, 3, 4, there are two 0 to 1 transitions. How many? 0 to 1. How many? There are 2 number of 0 to 1 transitions which means that we

define activity factor as equals to 2 by 4 which equals to 0.5. If there are 3 such cycles, it will be 3 by 5. If there are 4 such cycles, it will be 1 right? And then 4 by 4 is equals to 1. So activity factor is that.

(Refer Slide Time: 7:23)



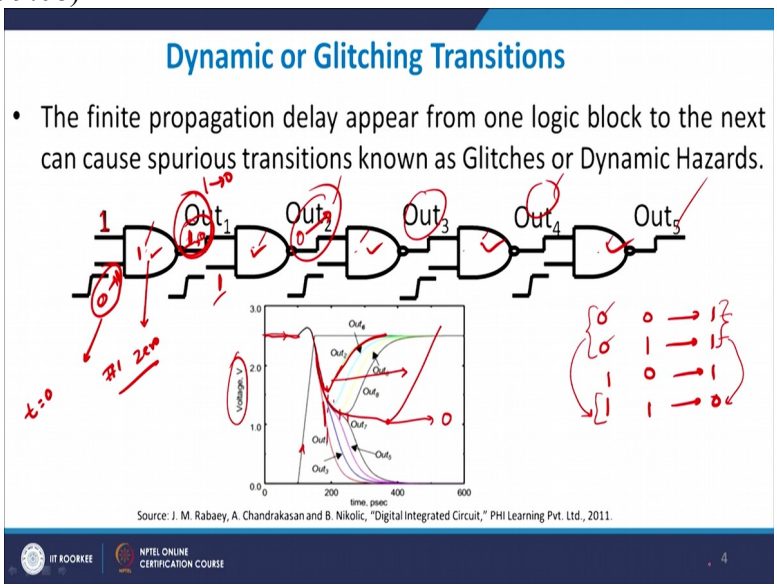
Now activity factor as I discussed with you, we will have two components, static component which is a function of the network topology. This is what I was talking about. So depending upon is it NAND to, NOR to logical or whatever logic you are using, you will have fixed number of 0 to 1 transitions for certain number of input clock period. That will determine your alpha 0 to 1 transitions. Now dynamic component which is the function of timing behaviour of the circuits.

So dynamic behaviour, dynamic component, it depends upon the timing behaviour, which means that say your clock is there but your, the input data train is there but the input data train is not same for both the inputs. There is some delay between them. Then there will be some glitches which we will discuss later on about which also alpha remains. But primarily, we will be looking into this 1st thing which is basically your static component, right?

We also define one important point that see, as I discussed with you, how do you calculate the probability? See, the issue is that the probability or alpha going from 0 to 1 is defined as the probability that one state is 0 and another state is 1. Of course, you can understand why. Because 0 to 1 means the initial state should be 0 and the final state should be equals to 1.

So if P_0 is the probability then the output will be in 0 state in one cycle and P_1 is the probability that the output will be in one state in the next cycle. Then the alpha 0 to 1 is also defined as P_0 into P_1, product of the 2 multiplications and therefore P_1 can be written as 1 minus P_0 because either you can have 0 or 1. Therefore 1 minus P_0 is the effective value of alpha 0 to 1.

(Refer Slide Time: 9:08)



Now let me come to dynamic or glitches, we will come back to this concept of alpha 0 to 1 later on. At this stage, you just have to keep in mind that this is basically the activity factor whose value is between 0 and 1 right? 1 is the highest value and which is basically a clock and it will go on decreasing depending on the topology which you are using it. C_L is the effective load which is visible to you at the end of the, at the combinational logic block, V_{DD} is the applied source voltage, voltage power supply and f is the frequency of operation.

Let us look at this concept of glitching and I will explain to you how does glitching affect your overall delay. Now, these are NAND gates here. So I have got 1, 2, 3, 4, 5 NAND gates and I am giving an input, 0 to 1 at this particular point, 0 to 1. So if you look at the NAND gate, its typical truth table will look like this, right? So suppose it is 1 right and here input goes from 0 to 1 in the input side. So initially when it is 1, 0, output will be equals to 1 and then when it goes to 1, this will glitch, this will go back to 0, right.

But this glitching takes place equally across the path. So you see, I have plotted here, output voltage here, the voltage at these particular points, out 1, out 2, out 3, out 4 versus the time and

this is my, so this is my input here, right and this is my output here. So whenever my input was low, input was low means one of them is, so whenever the input is low, means either this or this, the output is 11 and therefore output is latched to a high-value.

Input goes high and your, one of the other input was already high which means that I am going from this state to this state. So I am going from output 1 to output equals to 0. But the whole issue is this would have been true that all of the output will go to 0 together provided these did not have any intrinsic delay, right? If these did not have any delay, then this going from 1 to 0 will imply that this will happen instantaneously as 0 to 1 transition is taking place in input side. So out 1 transition of 1 to 0 will take place instantaneously as 0 to 1 transition in input side provided NAND 1, number 1 has got 0 delay. But this is not true.

And therefore, this 1 to 0 or 0 will appear in the output side of 1 only after a finite delay which is equal to the delay of this NAND gate. So let us see how does it influence my output. So you see, the output 1 goes to 0. You see. The output 1 goes to 0. This curve is basically output equals to 1 goes to 0. Why it goes to 0? Because when my input goes from 0 to 1, the output goes from 1 to 0. Now what has happened is, whenever my out, so you see, this has already gone to 1 right? Till the time when this output was equal to 0, till the time when it was, it came to 0 because initially you had 1 and then 0. 1 0 will give you 1 here.

So initially, you had 1 1 right? 1 1 means out 2 was start to falling down. So look at this, out 2 is the green one. Out 2 started to fall down. Why? Because at some value, suppose this is you did this at T equals to say, 0 milliseconds right and this has got a delay of 1 millisecond. So till 1 millisecond, this output will be equals to 1. So output will be 1 means this is 1, this is 1, this is 1, this is 1 implying that this will be always 0 and that is the reason, you see the output starts to fall to this value. It comes towards, it tries to go towards 0 but after 1 millisecond, this 0 appears.

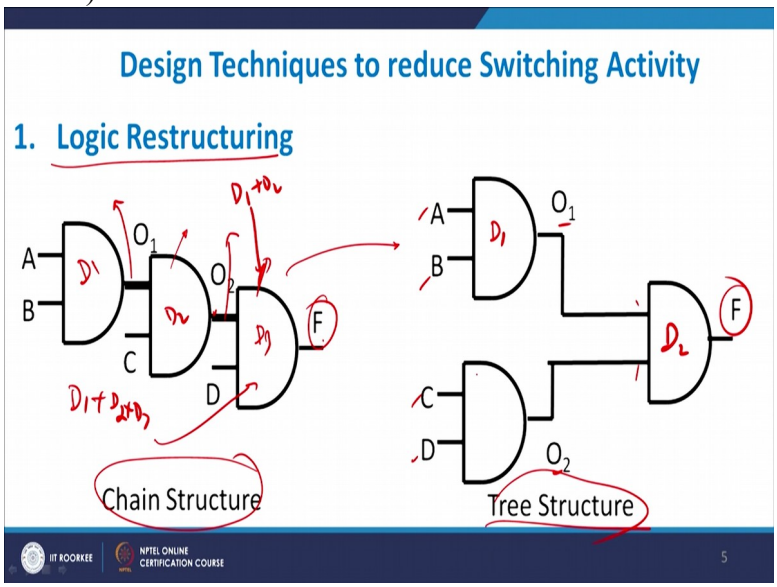
Why? Because you know how, you have a transition taking place in the output side. So out1 goes from 1 to 0. 1 to 0 primarily means that out 2 will go from 0 to 1. So now you see, the out 2 goes to 0 to 1. So you see, so similarly and so on and so forth, if you go in this direction, out 2, out 6, out 4, out 6 and out 8, all will first show a drop and then will go on increasing and as you can see in time domain, they are shifting this wards because this delay plus this delay plus this delay plus this delay plus this delay is all adding up.

The delays get added up and therefore the time till which it goes to 0 is actually shifting towards the right, right? And this is known as the dynamic hazard or a glitch. But ideally, so ideally if this chain would have been very long, a time would have come that I would have actually seen this going down like this and then increasing. So even if it is less than V_{DD} by 2, I would have seen it as 0 in reality whereas it should have actually gone to 1, right.

This is known as the dynamic transition or a glitching transition in a CMOS logic and that is one of the major important problem areas of glitching. So what the people do is something like this, that they try to do what is known as logic restructuring. So, how to reduce therefore the switching activity, right? Switching activity, remember was actually sorry, switching activity is actually equals to 0 to 1 transition. Now if you go back to 0 to 1 transitions, you see 0 to so what you are doing here, you are doing a 1 to 0 here, you are doing a 0 to 1 here, you are doing 1 to 0 here and 0 to 1, right.

So you are allowing larger number of 0 to 1 transitions and therefore power dissipation goes up, right. So this glitch helps you to increase the power dissipation, not a good idea right.

(Refer Slide Time: 15:20)



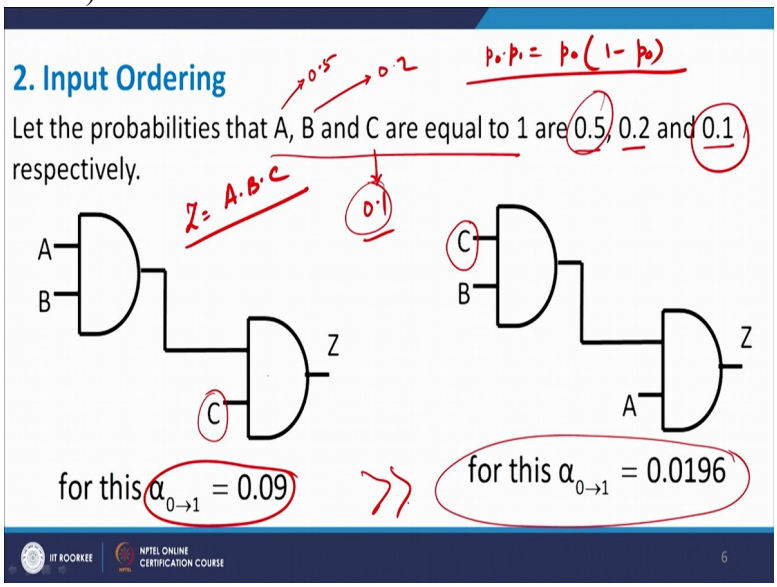
And therefore what we try to do is that we try to normalize the glitches or we reduce the glitches. How do we do it? By simply making logic restructuring and try to make it as symmetrical as possible, right? And rather than a chain structure, you do a tree structure. I will explain to you

how does it help you. In a chain as I discussed with you just now, you have to, so O_1 gate which is a AND gate has to wait till the delay of the 1st gate for this to evaluate from value of O_2.

Similarly, this has to wait, so O_2 has to wait till D_1 plus D_2 till this comes out and F has to wait till D_1 plus D_2 plus D_3. Okay, this D3 is the delay. You got the point? And therefore, these internal blocks will be glitching. And therefore, they will add to the power, total power, glitching power, right? So what you do? You convert this chain structure, this is known as the chain structure to what is known as the tree structure. The tree structure is the most symmetrical in nature which means that whenever AB is available to you and CD is available to you, O_0 and O_1 appear together and therefore this appear together and therefore F appears at exactly the same instant of time, right.

Overall delay also reduces because now you will have a overall delay of D_1 plus D_2. Initially, you had D_1 plus D_2 plus D_3. So the delay was much larger, frequency of operation will reduce, right. And you have a logic restructuring here which is basically a tree structure here and gives you a much lower logic, lower sort of activity here.

(Refer Slide Time: 16:58)



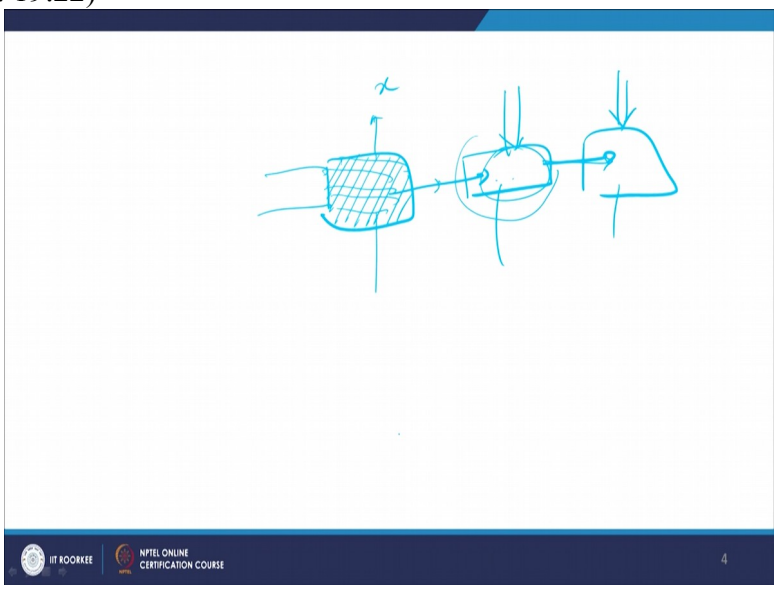
Another methodology which people adopted was what is known as the input reordering. Input reordering means that try to, so let us suppose that the probability of A, B, C equal to 1 are 0.5, 0.2 and 0.1. So A is basically 0.5, B is 0.2 and C is 0.1, the probability that they are equal to 1. Now if this C is inserted, so I have 2 logics. So if you look very carefully, Z is equals to A dot B

dot C, right? So it is a NAND logic which I am trying to realize. Sorry, AND logic, for 3 input AND logic. So that is why, if you are using a 3 input, we still can have, you can also have a 3 input AND gate but I am using a 2 input and gate, two 2 input AND Gates.

So what I do? I put A, B here and then C here. But C has got a probability of, the least probability of going to 1 right? In that case, if you try to find out alpha 0 to 1 right, from that initial P0 multiplied by P1 comes out to be equals to P0 into 1 minus P0, right? Similarly, you know what the values are. You can put the values here and you can get the value of alpha 0 to 1. But there is another method that the least value which you get, the least probability if you shift it to the earliest one and the most probability, if you shift to the later stage, you get alpha to be equals to 0.196 which is much lower than this value which means that, those inputs which have the highest probability of switching, try to put them closest to the output.

We have already discussed this point when we were discussing with you the propagation delay. Now in case of input reordering also, or input ordering also, you have to ensure that highest transition inputs which gives you highest probability of transition, should be delayed and should be placed at the last most gate for the consideration. As you can see, you have approximately 10 times decrease in the value of your alpha. And therefore, your dynamic power dissipation also goes down. Another methodology is time multiplexing the sources.

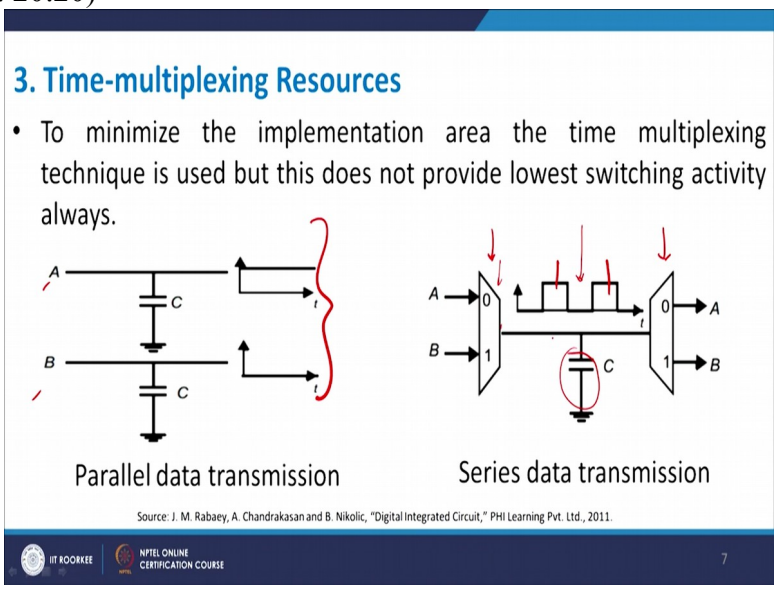
(Refer Slide Time: 19:22)



This is quite interesting that, what we try to do is that let us suppose I have 2 modules working together but, let us suppose I have got let me show it to you in this manner that I have got 2 modules working together and I have a block here and therefore they are talking with respect to each other. Now these are all driven by let us suppose the clock or even the data. Now this is behaving in such a manner that they are getting a certain intrinsic delay, say X, right. Now when the, so now you have fed an input here.

This is processing right? This is processing. Similarly, you do have some data, so this whole process give you a processed output, goes to this combinational logical block and then you forward it to this block. So what, what the concept is that till the time this is being processed, this 1st block, right, you can do some processing of the internal blocks as well. So that is what is known as the time multiplexed resource allocation. So you need to allocate resources here and try to make it lowest switching activity in this case.

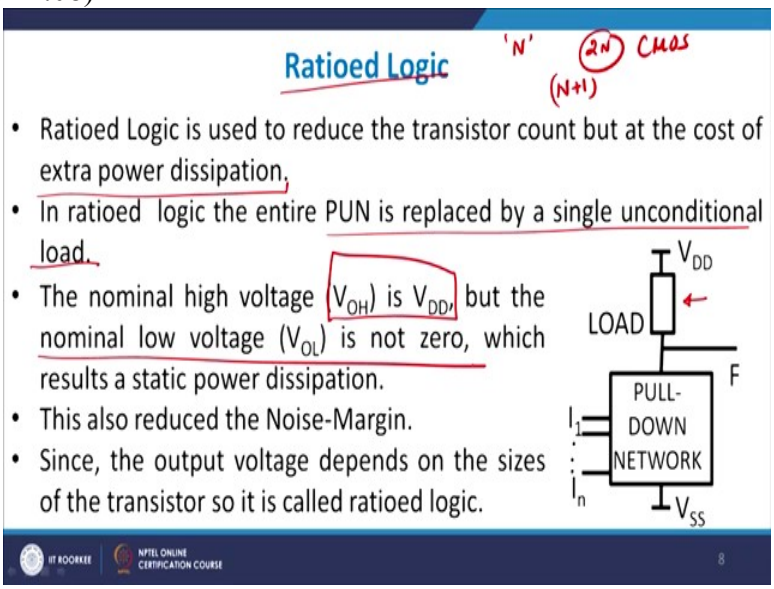
(Refer Slide Time: 20:20)



I will give you an example. For example, you have a parallel data coming into picture from A and B right and this C is the data. So I have a Mux here and I have a Demux here and this Mux serializes the data and this deserializes the data. So this is basically a serial deserialiser concept and you do have these output coming into picture here. So if we use such type of mechanism which is this one, then you allow for the capacitance to get charged and discharged and there will be sort of a talking between A and B.

As a result, there will be heavy glitches in this particular point. Whereas, in this case, you will have minimum glitch because there is only one line which is carrying the data from this Mux to this Demux. And therefore, this will give you a much lower power dissipation as compared to the previous case.

(Refer Slide Time: 21:08)



This takes care of approximately our understanding of complementary logic to larger extent and how the complementary logics can be used for a least delay and least power in a very sort of first-hand application area. We come to another one where it is known as the ratioed logic and ratioed logic primarily means that this logic has got lower transistor count as the previous one as a consideration. So if you remember, in a CMOS, P or CMOS architecture, you require for an N gate CMOS, for a N input CMOS, you require 2N Gates for a CMOS right?

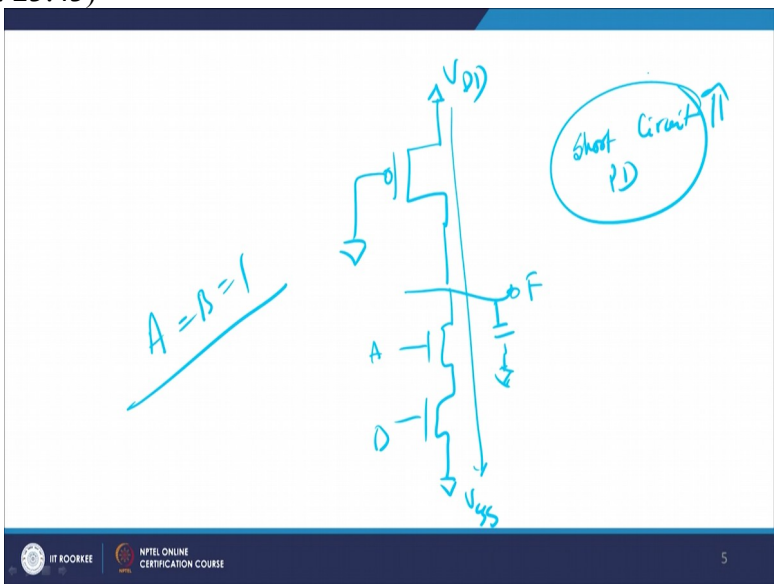
For a ratioed logic, it is just N plus 1. So if you are using a, for example a 4 input will be an AND gate or whatever gate, then for a CMOS logic input, you require 8 transistors. Whereas in this case, you just require 5. So the transistor count reduces, the area reduces. The cost you pay for it is higher power dissipation. So they have got extra power dissipation. I will explain to you why is it that. So what we do is, in a ratioed logic, what we do is we always keep the pull up network as a single PMOS device which is always on, let us suppose.

So what I do? I have a load here, the load which is given here in this figure. This is basically a PMOS and this PMOS is always on means its gate is always grounded. So that is what I am

saying that PUN is replaced by a single unconditional load right by a single PMOS and having exactly always on state which means that obviously, your output voltage will always be latched to V_{DD} because by that definition, till even a complementary CMOS. So therefore, my V_{OH} which is output high will be equals to V_{DD}, no problem.

But the problem is that your low voltage will not be equals to 0. Low voltage will not be equals to 0. It will depend upon the type of inputs you have given here right. Let us suppose you have given two transistors in series right and you have given 0 1, then 1 will be cut off and therefore you will never get to 0 right. Is it okay? But there is a problem here and therefore, you will have I will explain to you why do you will have static noise margin or you will have static power dissipation, I will explain to you.

(Refer Slide Time: 23:43)



See, for example, if you are, this is your PMOS right and a PMOS has been switched on. So it is grounded. So it is always on. And now I have got let us suppose a AND gate. So I have got A, B right and this is my F and this is C_L here. This is V_{DD} and this is your V_{SS}. Now if A and B are both equals to 1, then this will be switched on. But when this is switched on, you automatically have what? You have a direct path between V_{DD} and ground. So your short-circuit power dissipation, short-circuit is quite high, PD.

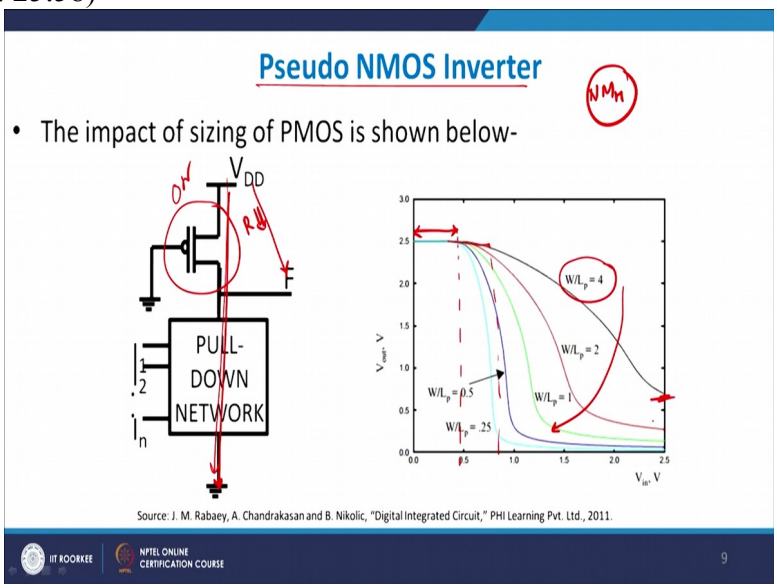
Your short-circuit powered situation is quite high. Why? Because now you have a direct path between V_{DD} and V_{SS}. So though the logic will give you a good value but otherwise, it will not

give you a good value in real sense. And that is the problem with static noise margins, with sorry, with the static power dissipation. 2nd thing is, we will not discuss in detail but it will reduce the noise margin as well. So pseudo-NMOS logic or a pseudo-logic it is also referred to as a pseudo-NMOS logic, also known as the ratioed logic will have a lower noise margin, right.

And the reason is that your pull up network and pull down networks are not equally strong. For example, your pull up network might be very weak because you are using a low value of W/L because you want the RL to be very high, load to be very high, you want W/L to be small but once you make W/L small, you are making a pull up weak. When pull up is weak, your high noise margin reduces, right.

And that is the major area of concern for ratioed logic, right and that is the problem area this thing. Now, therefore since the output voltage depends on the size of the transistors, so it is called a ratioed logic. Why is known as the ratioed logic? Because the output value of voltage will depend upon the size of the transistor. Higher the size better the transistor will be able to make the output voltage go to ground and lower the size, difficult it will be go to the ground. So therefore, this is known as the ratioed logic as I discussed with you. Also, therefore also this is also referred to as a pseudo-NMOS inverter as I discussed with you.

(Refer Slide Time: 25:58)



So what I do? I PMOS, I make it grounded. So this is always on state right and then I have a pull up network with me. Now as you can see here, my pull up network is not strong because I have

only one PMOS here and therefore, your noise margin which is high noise margin, is very small. You see, it is just this much, almost, approximately this much for W by L P_5, is this much which means that, if the noise flow or the noise is typically low, even then this will not be rejected by the inverter at high values.

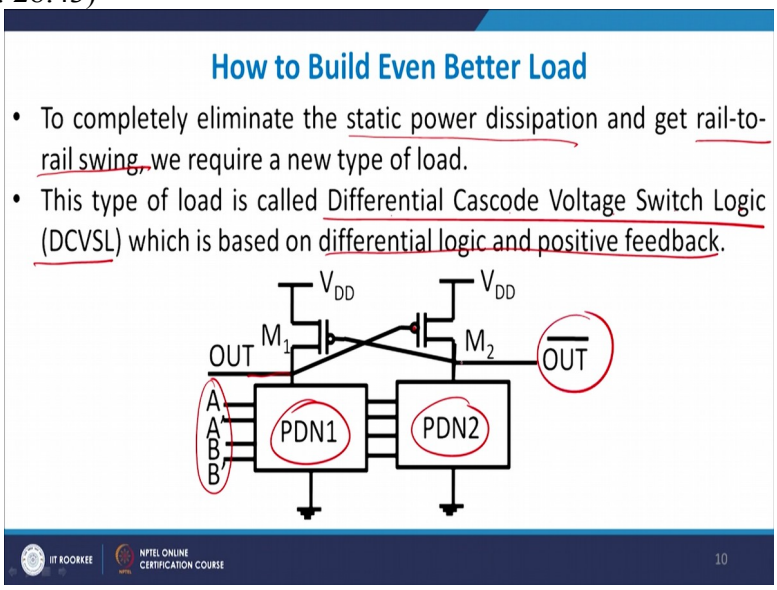
Now as you can see here, so this is W by L of PMOS. So as the W by L of PMOS is increased, increased means R value goes on reducing and you can easily have V_{DD} appear as F and therefore you see your noise margin goes on increasing. So the noise margin for W by L P_4 is approximately this much, whereas it goes on reducing as you go on reducing the value of W by L of P. So once the aspect ratio of your pull up transistor goes reducing, your noise margin reduces.

Not only noise margin reduces, you also have a problem that, for example if you are saying a W by L P equals to 4, your output when you are input is high should so when you are input is high, output should go to ideally go to 0 because let us suppose, it is a two input AND gate, then 11 will give you 0. So but it does not go to 0. Why it does not go to 0? The reason is, you always have a direct path between V_{DD} and this V_{SS}. And higher the value of W by L of PMOS, lower is resistance and that larger current will be flowing through V_{DD} and V_{SS}.

As a result, it is never latched to 0 in this case. So at the cost of higher noise margin, my output does not go to the 0 value where it should go. And therefore, when you cascade a ratioed NMOS inverter to another one where you expect that the, so if you are cascading this inverter with another inverter and that inverter is expecting a 0 from this inverter, but in reality, it is not giving me 0, it is giving me 0.5, 0.6, 0.7 voltage, there will be problem as far as its operation is concerned. And that is the major concern of a pseudo-NMOS logic.

There are 2 concerns. One concern is that your output does not even go to 0. It goes to 1 very well, it does not go to 0 properly because of the always on, this PMOS is always on right. And the 2nd property is its static power dissipations are very high. So its overall power dissipations are very, very high.

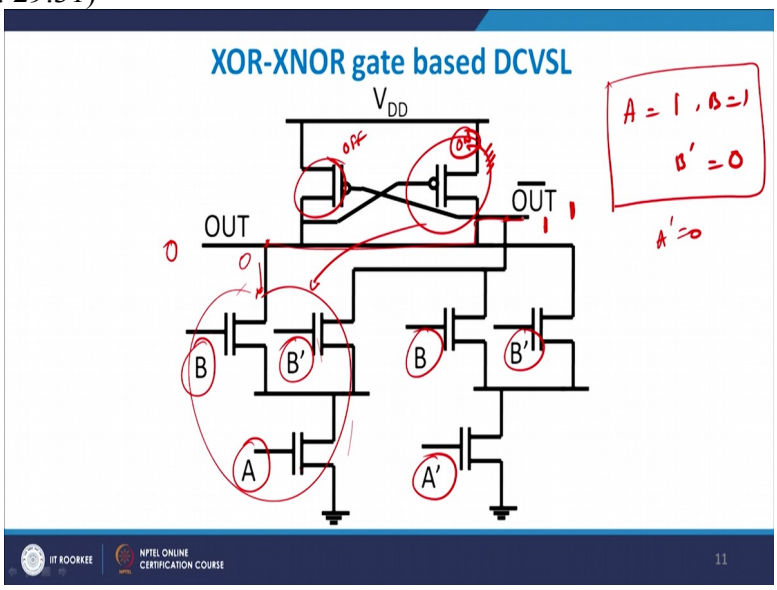
(Refer Slide Time: 28:43)



So let me see how can you build a better load or maybe a good one. And this is known as what is known as a Differential Cascoded Voltage Switch Logic, DCVSL. Differential Cascoded Voltage Switch Logic, also known as DCVSL, based on differential logic and positive feedback. So what you do is very simple. You have a PDN and you took a PDN2 which is just complementary of PDN1 and you apply both A and its, so if you have two signals, you also apply its complementary signal and what it does?

It totally removes your static power dissipation and you automatically get a rail to rail swing and I will show you how it works out. So you see this PMOS is connected, the output here is connected to the gate of PMOS M_2 and out bar is connected to M_1 gate here.

(Refer Slide Time: 29:31)



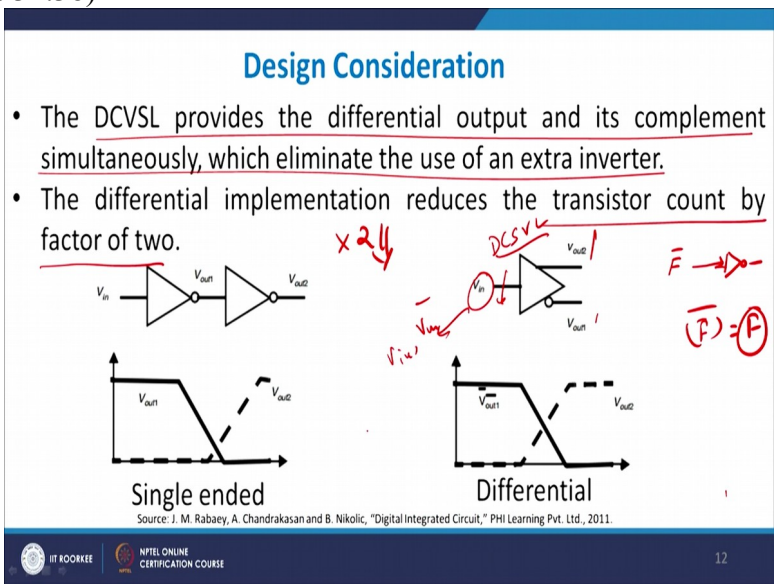
Let us look at the functionality of a DCVSL XOR-XNOR gate. Now you see, in this case if A equals to 1 right, B equals to 1 and therefore B bar will be equals to 0. In such a scenario, what I get is that I have, so what I did was, I told you PDN should be complementary in the sense that if you apply A signal here, you should apply A bar signal to PDN2. If you apply B signal here, you should apply B bar here. If it is B bar, you apply B here, right. So let us suppose A equals to 1, B equals to 1, B bar equals to 0.

When A equals to 1 and B equals to 1, this out goes to 0, so out of goes to 0. In this case, A equals to 1 primarily meaning A bar equals to 0. 0 means this is cut-off. When this is cut-off, means this goes to high. So this goes to 1. This goes to 1 primarily means that this PMOS is goes to off and this is on. Once this is on, so this transistor works with this module to give you an output here. So out becomes equal to 0 and out bar equals to 1. So in a DCVSL logic, I insert differential signals, I also get differential signals back which is 0 and 1 here, right.

So I input differential signals, A and A bar, A and B bar and I also get differential signals out from out and out bar. This is what we get in DCVSL logic but the idea there is that there is no direct path between V_{DD} and ground at any point of time which was there in a pseudo-NMOS logic. In a pseudo-NMOS logic, you did have a direct path from V_{DD} to ground. In this case, you do not have any direct path. So you do have a simple, so you see, when this is on and when this is on, you do not have a direct path.

So you see this is goes like this and this is the output which you see. So this is on right and this is switched on here. This is switched on, this is switched on and therefore this output goes to 0 and therefore it almost out bar equals to 1. And therefore, this gives you a very, very low static power dissipation right.

(Refer Slide Time: 31:38)

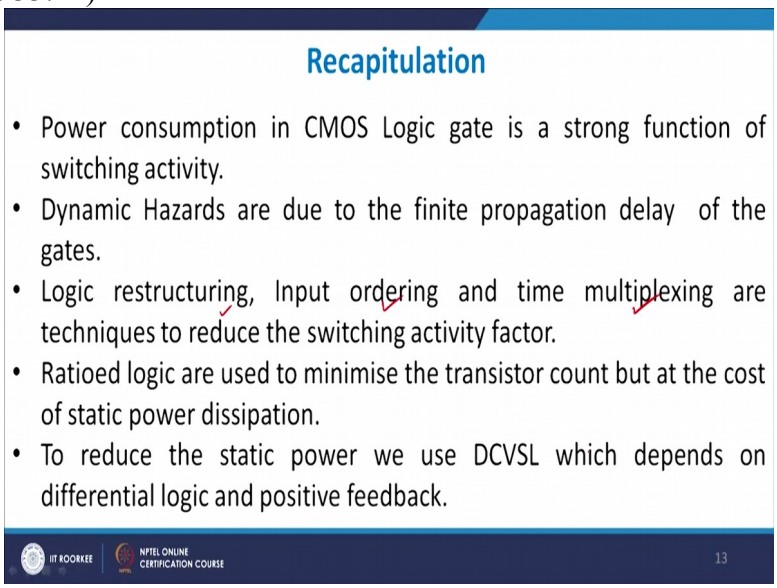


So therefore, DCVSL provides a differential output and its complement and therefore it eliminates the use of external resistor. So you remember, when we were using a ratioed logic or when we were using a standard CMOS complementary logic, my output will always be complementary. Remember, by virtue of the fact that it is an inverter. So you have to put an extra static inverter to complement it back to its original form. So I generally get F bar and therefore when you pass it through a static inverter, I get F bar of bar and therefore I get F.

So I am using one more static inverter to achieve the signal. In this case, we do not do that. In this case, we already get both the signals and its complementary together. The differential implementation reduces transistor count by a factor of two. So you get a 2 times reduction in the transistor count in this case. Now, so I get V_{out}. So this is a V_{out1}, V_{out2}, V_{out1}, V_{out2}. This is single ended and this is a differential operation as you can see V_{in} I do and I get V_{out2}. So this is basically your DCVSL but the problem is that, V_{in} should also be Vin and you should also give Vin bar, the complementary forms should also be given right and that gives you quite a good idea about this whole thing.

So let me recapitulate up this module what we have done till now. We have learned that the power consumption is a very, very strong influence on the switching activity. Switching activity plays an important role in determining the power consumption.

(Refer Slide Time: 33:22)



The power consumption depends upon C_L the load, V_{DD} the applied voltage, and f the frequency of operation. It also depends upon the topology which you have with you and also depends upon dynamic glitches. So the methodology we adopted for reducing dynamic glitches are logic restructuring, right, logic restructuring, input ordering and time multiplexing some of the techniques because this will reduce your dynamic hazards.

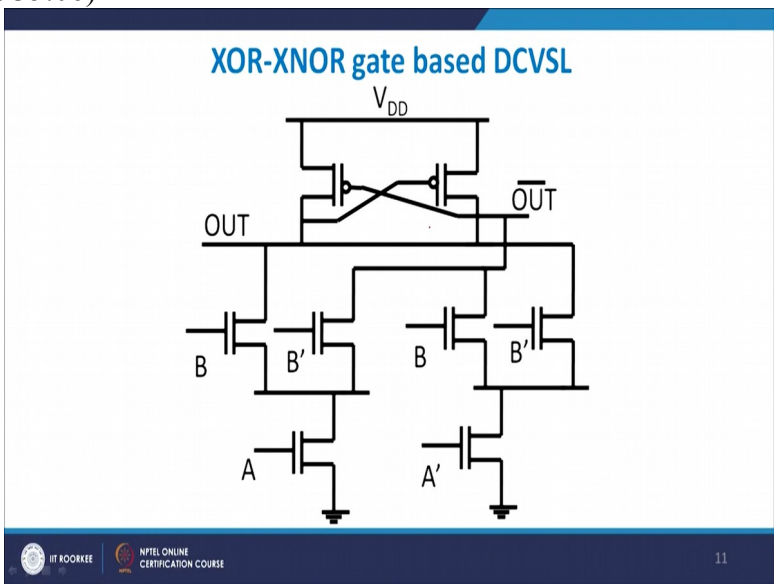
And dynamic hazard appear because of finite propagation delay of the Gates, right and that gives you an important one. To remove it people went for ratioed logic, in which ratioed logic was what? That you replace the pull up network by a single PMOS device which is normally on and you make it always on because you ground the gate of the PMOS and then you try to get the output. But the problem there was that you have two problems.

One was that you did not allow the output to go to fully to ground if your load was having a large W by L ratio because now the load will try to pull up the voltage towards itself and therefore you will not allow it to go to ground and that is the major problem area and therefore you and therefore, the 2nd problem which came into picture was that the noise margins were reduced

drastically, specially the low noise margins were reduced drastically. High noise margins will depend upon the W by L ratios of the PMOS which you take.

So higher the W by L ratio, larger will be the static noise margin in this case. Now to reduce static power and to get a differential output, we did DCVSL which is Differential Cascoded Voltage Logic and in which case, you use a differential input, you do a positive feedback and then you do this thing. Just to give you a brief insight what positive feedback helps you to do what I will tell you.

(Refer Slide Time: 35:06)



It tries to reduce the delay drastically. See when you have a positive feedback and suppose this is 0 and this is basically your 1 here, this 1, so this 0 will try to switch on this device very fast, right and therefore the overall delay of a DCVSL is reduced drastically. So since this is a positive feedback, you do not have to wait till a particular time for the output to appear and therefore suddenly when it goes to 0, because by application of A equal to B equals to 1 and B bar equals to 0, I automatically get a much lower profile here, right?

So we have understood this whole logic here. We have therefore taken care of ratioed logic, DCVSL, understood what is power dissipation and we also know what is propagation delay. So by this, end of this module, you as a deliverable, you should be able to deliver the logic, you

should be able to design a logic in a much better manner using CMOS technology, using pseudo NMOS technology and using a DCVSL.

And please try to do it as a sort of a home assignment once you go back. Good book to look into is basically, there is a good book by Digital integrated circuits by a designer's perspective by Chandrakasan, Rabaey and Nikolic. I will recommend that you please go through that book for at least the digital part. Analog part you will not get there but the digital part of this whole course structure is very well defined in that case. Thank you. Thank you for your kind, patient hearing.